DNSOP WG
INTERNET DRAFT
Category: I-D

D. Massey, UCLA
T. Lehman, ISI
E. Lewis, NAI Labs
October 20, 1999

DNSSEC Implementation in the CAIRN Testbed.
<draft-ietf-dnsop-dnsseccairn-00.txt>

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet- Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

Comments should be sent to the authors or the DNSOP WG mailing list dnsop@cafax.se.

This draft expires on April 20, 2000.

Abstract

This draft describes the operational and administrative considerations encountered as part of the migration from a traditional Domain Name Service (DNS) infrastructure to one based on DNS Security (DNSSEC). The information presented in this draft is based on the implementation of DNSSEC in the Collaborative Advanced Inter-agency Research Network (CAIRN) testbed. CAIRN is a DARPA funded testbed used by Government, University, and Commercial researchers to conduct Internet Protocol (IP) network based research. Additional information on the CAIRN testbed is available at http://www.cairn.net. Details on the DNSSEC implementation in CAIRN can be found at http://www.cairn.net/DNSSEC.

It is envisioned that the information presented in this draft can be used as a starting reference for sites interested in deploying DNSSEC.

It is also hoped that reporting on the experience of CAIRN DNSSEC implementation will provide a forum for general discussions regarding DNSSEC implementation.

TABLE OF CONTENTS

1 INTRODUCTION

The DNS Security Extensions (RFC 2535) provide a mechanism for authenticating DNS responses. Authentication is accomplished by including digital signatures in the response sent by DNSSEC aware name servers. RFC 2535 describes the DNSSEC extensions in detail. Deploying DNSSEC requires new operational duties related to key generation, zone file signing, and key management. This draft describes the operational and administrative changes encountered when migrating from traditional DNS to DNSSEC.

Section 2 describes the DNSSEC architecture of CAIRN as well as the objectives for the DNSSEC implementation. Section 3 describes the procedures and issues associated with key generation. Section 4 describes the procedures and issues associated with zone file signing. Section 5 describes the procedures and issues associated for submitting keys to a parent zone for signature and creating a secure sub-zone. Section 6 presents some operational scenarios. Section 7 concludes with a discussion of incremental deployment and

mechanisms to facilitate implementation in the general Internet. Section 8 discusses open issues and areas of interest for future testing within CAIRN.

The operational changes described in this draft are based on experience gained by deploying DNSSEC in the CAIRN network. It is expected that the operational procedures described here will change as the DNSSEC specification evolves and more operational experience is gained.

Feedback from other sites who have deployed (or are planning to deploy) DNSSEC is strongly encouraged.

1.1 Misc. Notes

The deployment described here does not address the use of dynamic updates.

This draft includes the use NXT records for purely pragmatic reasons. The use of NXT records is still being debated and some feel strongly that NXT records should not be included in DNSSEC. NXT records are used here since the current DNSSEC specification uses NXT records and the DNSSEC tools distributed with BIND generate NXT records.

This draft will focus on the operational and implementation considerations associated with the DNSSEC implementation in CAIRN. It will not attempt to include detailed background on DNSSEC, except for that necessary to discuss the issues contained in this draft. The reader is referred to other RFCs and Internet Drafts for more detailed treatment of DNSSEC in general. These include but are not limited to:

Domain Name System Security Extensions (RFC 2535)
Handling of DNS zone signing keys <draft-ietf-dnsop-keyhand-00.txt>
DNS Security Operational Considerations (RFC 2541)

2 ARCHITECTURE AND IMPLEMENTATION OBJECTIVES

2.1 Architecture

The CAIRN testbed has administrative control via delegation from the root name servers for the following zones: cairn.net, 173.140.addr.arpa, and a.1.e.f.f.3.IP6.INT. The root servers delegate the cairn.net zone to the following name servers: lila.east.isi.edu, ns.isi.edu, and flag.ep.net. The master name server is lila.east.isi.edu. The other two name servers are slaves. At this time only the cairn.net zone and other sub-zones within that hierarchy are implemented as secure zones. There are approximately 30 delegated sub-zones within cairn.net. Some

of these delegations are implemented as secure sub-zones, others are not secure.

There are also numerous secondary name servers for the cairn.net zone.

Since this is a research network the the configuration of the delegated sub-zones and their secure status changes depending on current network activity. For this reason a detailed description of the current CAIRN DNS architecture is not presented here. However, the interested reader is encouraged visit
http://www.cairn.net/DNSSEC
for the latest configuration and implementation status.
The DNSSEC implementation in CAIRN is based on the versions of BIND available via Internet Software Consortium (ISC).

2.2 Implementation Objectives

Detailed descriptions of the administrative tasks associated with creating and maintaining a secure zone are presented in subsequent sections of this draft. These tasks are organized as shown below.

i) generation of "zone keys" (described in Section 3)
ii) signing of the zone file (described in Section 4)
iii) signing zone keys and administering delegations
(described in Section 5)

The main administrative burden is expected to be the activities associated with item iii above. Upon generation of zone keys, a secure zone must have its public key signed by an appropriate authority. The appropriate authority is usually the parent or delegation point for the zone. This requirement will impose burdens on both the zone which is configuring itself for DNSSEC, as well as the parent for that zone.

The implementation objectives for DNSSEC in CAIRN are as follows:

I) Minimize the impact on a "parent" zone file in terms of modifications required due to a change in the secure status or configuration of a sub-zone.

The desire is that a "parent" zone file should not have to be modified if a delegation point changes its secure status, generates new keys, or makes any other changes to its configuration which would not require a parent zone file modification in the traditional DNS implementations. It is felt that the interaction between a parent and child should be constrained to off-line signature of the child's keys by the parent. Changes in child's keys should not require a resigning of the parent's zone file. In order to

accomplish this, the parent zone file will contain the same information it does today regarding delegation points. This limits the information contained in the parent zone file to the sub-zone name and the name servers which serve that zone. Key information for sub-zones would not be included in the parent zone file under this scenario.

II) Provide mechanisms which allow for non-contiguous secure zone space.

It is expected that initial implementations of DNSSEC will result in a DNS hierarchy which has a mix of secure and non secure zones. In addition, it is further anticipated that within the DNS tree, the secure zone space will be be non-contiguous. It is desired that there be mechanisms that allow an "island" of secure DNS hierarchy to exist and still be able to provide secure name resolution for those name servers and application resolvers which are adequately configured. It is envisioned that the "parent" signing hierarchy

must be allowed to deviate from the general DNS hierarchy to accommodate this.
The approach taken for the CAIRN DNSSEC implementation to accomplish these two objectives is presented in subsequent sections of this draft. Specific example scenarios are presented in section 6.

3 KEY GENERATION AND MANAGEMENT

Prior to DNSSEC, there were no DNS keys to manage. DNSSEC adds new requirements for creating and managing zone keys. DNSSEC authentication is based on the use of public/private zone keys.
A zone creates public/private key pairs. The private key is known only by the zone and is used to sign the zone records. The public key is made widely available by placing it in a KEY record. Resolvers who trust the public key can verify signatures and authenticate records received from the zone.

Generating new keys is a simple process that can be done with standard tools such as the dnskeygen tool included with BIND distribution. Keys may be generated for several different algorithms. In CAIRN, a DSA key is mandatory and must be created and an RSA key is optional. A zone decides which algorithms it will support and generates the corresponding key pairs.

A crucial task for a DNSSEC zone is keeping private keys private and insuring that key pairs expire in a reasonable time. An attacker can forge DNSSEC data if he either gains access to a private key or reconstructs a private key via cryptoanalysis techniques. To provide reasonable security, each public/private key pair should

periodically expire and be replaced. The recommended lifetime for key pair is dependent on both the key size and the algorithm. The key pair should be immediately discarded if the private key is believed to have been compromised.

Once a site has chosen its algorithms and generated its zone key pairs, the public version of each key is placed in a KEY record. The set of KEY records must be signed and then placed in the zone's master file. Note that common tools such as dnskeygen automatically create the KEY record.

CAIRN generates the following keys for the cairn.net zone:

cairn.net master DSA key - this key is expected to remain unchanged for an extended period of time. It will be generated using the DSA algorithm. Its operational purpose is to sign other keys. The public portion of this key is what will be distributed to all other name servers as the top level key. This key represents the key that in a true "production" environment would require the most strict security measures to ensure it was not compromised.

cairn.net master RSA key - similar to the master DSA key.

cairn.net zone signing key 1 - this key will be changed more frequently than the master key. It will be generated using the

DSA algorithm. Its operational purpose is to sign the cairn.net zone file.

cairn.net zone signing key 2 - this key will be changed more frequently than the master key. It will be generated using the RSA algorithm. Its operational purpose is to sign the cairn.net zone file.

The CAIRN testbed uses the dnskeygen program which is included with the BIND distribution. The following examples shows how a DNSSEC key is generated for the cairn.net zone:

```
dnskeygen -D 768 -z -c -n cairn.net
   -D defines the key algorithm to be DSA
   -z indicates a zone key is being generated
   -c indicates the key cannot be used for encryption
   -n sets the keys name
```

The output of this program would be two files which contain the public and private parts of the key. The file names will be of the form:

```
Kcairn.net.+X+Y.key       -public portion of key
```

Kcairn.net.+X+Y.private   -private portion of key

where X indicates the algorithm type and Y is the footprint of the key.

Once the keys are generated, the public portion of the zone keys are included in the zone file in the form of a KEY record. A single file is created for this purpose which combines the contents of all the Kcairn.net.+X+Y.key files into a single file named cairn.net.keys. Its contents will be of the following form:

cairn.net. TTL IN KEY flags algorithm protocol key_value

cairn.net. TTL IN KEY flags algorithm protocol key_value

cairn.net. TTL IN KEY flags algorithm protocol key_value

Creating the key file is currently a manual process, but a script to automatically assemble a key file is being investigated.

The key generation and management procedures and decisions for sub-zones are expected to differ from those described above for the cairn.net zone. The CAIRN administrative philosophy is that sub-zones are should be free to determine the key algorithms, numbers, and lifetimes as required to meet their operational and administrative needs. However, due to implementation issues described in section 5, CAIRN currently allows only RSA and DSA keys.

Now that once the zone keys are generated, administrative activities can continue with signing of the zone file.

4 ZONE FILE SIGNING

Prior to DNSSEC, a change in the local DNS data required an administrator to edit resource records in the master file, update the SOA entry, and refresh the zone. DNSSEC adds two new operations: resigning the zone file and updating the NXT records.

After editing the resource records and updating the SOA, an administrator must sign the changed (or new) records with the zone keys and update the appropriate NXT records. Signing the changed records with the zone keys will produce the SIG records. A signed zone file should include one SIG record per zone key. The SIG record contains the digital signature and related information which can be used to authenticate the data.

The NXT records are also stored in the zone file. The NXT records in a zone create a chain of all of the literal owner names in that zone and provide a mechanism to indicate which the data is not

present in the zone file. NXT records require that the records in a zone file be placed in a canonical order. A NXT record specifies that there are no records between two ordered records in the zone file. The dnssigner tool provided with the BIND distribution automatically orders the zone file and generate the NXT records.

In order to create a signed zone file, a typical zone file is created. A portion of a cairn.net zone file may be as shown below:

```
_______
$ORIGIN cairn.net.
@       86400   IN      SOA     lila.east.isi.edu.
                                root.lila.east.isi.edu. (
                        1999101801      ;serial number
                        3600            ;refresh time (60 min)
                        300             ;retry time (5 min)
                        604800          ;expire time (1 week)
                        86400 )         ;minimum time (1 day)

@       86400   NS      lila.east.isi.edu.
@               NS      ns.isi.edu.
@               NS      flag.ep.net.

$INCLUDE /etc/named/keys/cairn.net.keys

localhost       86400   A       127.0.0.1
loghost                 CNAME   localhost.cairn.net.
loopback                CNAME   localhost.cairn.net.

;misc addresses
www                     A       38.245.76.15
loon                    A       38.245.76.29
.
.
.
______
```


The only unique item at this point is the inclusion of the cairn.net zone KEY records via use of the INCLUDE statement.

The CAIRN testbed uses the dnssigner program which is included with the BIND distribution to sign this zone file as shown below:

```
dnssigner -ess -zi cairn.net -zo s_cairn.net -ks cairn.net
          001 footprint cairn.net 003 footprint
```

-ess tells dnssigner to have each private key sign the other public keys. The objective here is to have the cairn.net master key sign the cairn.net zone signing

key 1 and key 2.

-zi identifies the unsigned input zone file

-zo specifies the name of the output file which will contain the SIG and NXT records

-ks indicates that there are multiple keys to sign the zone with. The actual keys are specified following this command line switch by identifying the algorithm and footprint. The files containing these keys must be present in the same location from which the dnssigner program is being run.

The output of the above command will be a zone file with SIG and NXT records. For the CAIRN configuration all records in the original unsigned zone file will be signed by the cairn.net zone signing key 1 and the cairn.net zone signing key 2. In addition the cairn.net zone signing key 1 and the cairn.net zone signing key 2 will be signed by the cairn.net master keys.

4.1 Zone File Signing Security Considerations

Signing the zone requires simultaneous access to both the private keys and the modified zone file. The zone file resides on the primary name server. However, keeping the private keys on the primary name server is not recommended. In an ideal scenario, the private keys are kept off-line. This adds the additional administrative burden of transferring the zone file to location of the private keys and then transferring the signed file back to the primary server.

Care must also be taken to insure that the signatures are given the proper expiration dates. A signature expiration date should be no later then the anticipated expiration date for the corresponding public key. It is expected that different keys will have different lifetimes.

5 SUB-DOMAIN ADMINISTRATION

Prior to DNSSEC, a zone only stored the name servers associated

with its sub-zones. Interaction between a zone and a sub-zone was required only if the sub-zone changed name server locations. DNSSEC requires a much higher degree of coordination between a zone and its sub-zones. The increased coordination is a result of key exchanges and key signing that must occur between a zone and its sub-zones. Keys change frequently, at least once a quarter in CAIRN, and key exchanges between a zone and is sub-zones must be

done in secure fashion. This secure key exchanges requires a new style of interaction between a zone and its sub-zone.

The CAIRN administrative philosophy is to minimize the interaction between zones and sub-zones as much as possible and to allow for the incremental deployment of DNSSEC. To accomplish this, a CAIRN zone stores only its own KEY records. A zone does not store any keys which belong to any of its sub-zones and does not generate any keys on behalf of any sub-zones. These rules insure that any change in a sub-zone never require a corresponding change in the parent's zone file. RFC 2535 requires one exception to this policy with respect to NULL keys for a unsecure sub-zone. Section 5.1 describes the CAIRN policy for unsecure sub-zones and the exception. Section 5.2 describes the policy for secure sub-zones. Incremental deployment issues and key chaining across an are discussed in sections 5.3 and 5.4, respectively.

5.1 Administration of Unsecure Sub-zones.

The CAIRN policy is that a zone is responsible for only its own KEY records. A zone is not responsible for holding or generating KEY records on behalf of any other zone. However to comply with RFC 2535, one exception must be made in this policy. If a sub-zone is completely unsecure, then the parent zone must generate a NULL DSA KEY and a NULL RSA KEY for its unsecure sub-zone. These NULL keys are stored in the parent zone. No communication with the sub-zone is required. The NULL KEYS should have no expiration dates and require no further administration by the parent (other than periodic signing which should automatically occur whenever other records in the parent's zone file are signed).

A sub-zone can switch to secure status by notifying its parent. Once a sub-zone notifies it parent, the parent removes the NULL keys and the parent assumes no further responsibility for the security status of the sub-zone. The CAIRN policy of not generating or storing KEYs for the sub-zone comes back into effect. A sub-zone can switch back to unsecure status by generating and storing its own NULL KEYs. By generating its own NULL KEYs, a sub-zone can change its own security status without requiring a change at the parent zone. If a domain switches to unsecure and has no plans to return to secure status, then the sub-domain may request the parent resume generating and storing NULL KEYs on behalf of the sub-domain, but this results in changes to the parent's zone file is discouraged.

As part of our experimentation we will be considering the best way to generate and store NULL keys. However, based on our activities to date there are concerns that requiring a parent to maintain null keys

and assume a degree of responsibility for a child domain for which it does not administer is not recommended. It is felt that this will impose too great a burden on a parent domain and will hinder an incremental implementation of DNSSEC in the Internet. We request further clarification on the NULL KEY in the DNSSEC specification and seek advice for handling the following three problem scenarios.

First, consider the problem of a secure sub-domain which later becomes unsecure. After some time, a secure sub-domain fails to send any KEY records to the parent for signing. The sub-domain has only KEY records with expired SIGs from the parent or the sub-domain has simply stopped using DNSSEC and has no KEYs or SIGs. Given the experimental research nature of CAIRN, this scenario will occur. Is it the responsibility of the parent to declare the sub-domain unsecure and generate new NULL keys on behalf of the sub-domain? What responsibilities does a parent have in insuring a secure sub-domain really is secure? If the parent has no responsibility, is a NULL key at the parent still a meaningful concept?

Second, consider a conflicting view of a sub-domain's security status. Zone sec.nosec.cairn.net is secure, but nosec.cairn.net had not implemented DNSSEC. To overcome this, the domain sec.nosec.cairn.net has its KEY set signed by cairn.net. However, later nosec.cairn.net implements DNSSEC and mistakenly assumes its sub-domains are unsecure. There are now valid keys for sec.nosec.cairn.net which are signed by cairn.net. There are also NULL keys for sec.nosec.cairn.net which are signed by its parent domain, nosec.cairn.net. How should resolvers treat the domain sec.nosec.cairn.net? If both a parent and sub-domain get decide whether the sub-domain is secure, who is right if they disagree?

Third, given the multiple key types available, which NULL keys must be generated by a parent? Suppose domain.cairn.net defines a local algorithm type of 253. Zone sub.domain.cairn.net also defined a DIFFERENT local algorithm type of 253. Should domain.cairn.net generate a NULL key for its child? If so, what does this NULL key mean given that the child does have a KEY for a type 253 algorithm? If the parent does not generate a NULL KEY for type 253, doesn't this incorrectly imply the sub-domain understands the parent's algorithm 253? How are NULL KEYs applied to local algorithms?

5.2 Administration of Secure Sub-Zones

A domain declares itself secure by notifying its parent and sending keys to the parent for signing. A domain is responsible for insuring that its KEY set is signed by its parent at the appropriate times. The domain should send new KEY sets whenever the SIG records from the parent expire or whenever the domain generates new KEY records. The parent may impose limitations on the frequency of KEY set signing and the domain must provide an emergency contact who can be reached if a key set is believed to be compromised.

Key signing is currently accomplished by an off-line exchange of keys. A sub-domain sends a set of fully specified KEY records to

the parent domain. The parent domain signs the KEY set, exactly as reported by the sub-domain. The sub-domain must insure that the KEY set has the appropriate TTL values and contains any NULL KEY records if desired. The parent domain simply signs whatever records its receives. The parent is strictly forbidden from making any modification to the KEY record set. The resulting SIG records are returned to the sub-domain. Once the SIG records are returned, the parent should discard the KEY set and SIG records. No changes are made to the parent's zone file.

After receiving the SIG records from its parent, a sub-domain must check each SIG record before adding the SIG record to the zone file. A sub-domain must discard a SIG record if the signature does not match the original KEY set or if the sub-domain does not implement the algorithm used to create the SIG. After the off-line KEY exchange and SIG verification is complete, the sub-domain stores its KEY record set and the accepted SIGs in its zone file.

Administrative issues related to NULL keys and multiple key algorithms are addressed by administrative policies. By administrative fiat, secure CAIRN domains are required to generate DSA zone keys. This insures that every sub-domain will receive at least one signature which it understands and can verify. CAIRN zones may also optionally chose to generate RSA zone keys. A zone which chooses not to generate an RSA key must generate a NULL KEY for the RSA algorithm. This implies that the KEY set sent to a parent must consist of either DSA and RSA key records or DSA key records and a NULL RSA KEY record. Until further clarifications in the DNSSEC specification are made, additional algorithms are not allowed (see section 6.4 for an example of the problem). CAIRN eventually hopes to allow any algorithm type, including locally defined algorithms.

5.3 Incremental Deployment Policies

Additional administrative problems are expected to arise during the incremental deployment phase of DNSSEC. In both CAIRN and the general Internet, not all domains can be expected to deploy DNSSEC simultaneously. Some domains will find their parent does not support DNSSEC and is unable to sign the KEY record set. Currently the net. domain is unable to sign the cairn.net. KEY set. Within CAIRN, a domain such as sec.nosec.cairn.net may deploy DNSSEC, but its parent domain nosec.cairn.net may choose not to deploy DNSSEC. CAIRN has adopted the administrative polices described below in order to handle these scenarios.

5.3.1 Signatures for cairn.net Keys

CAIRN maintains a master key and suggests that all CAIRN domains include this master key in their trusted key list. This universally trusted master key is used to sign the cairn.net zone keys. It is expected that master key will be kept highly secure and will change infrequently. The cairn.net zone keys expire on a quarterly basis. Secure resolvers should agree to trust any cairn.net KEY

records that are covered by a valid SIG from the master key. Any change in the master key would require reconfiguring the trusted key list of all CAIRN domains.

This policy is based on the DNSSEC approach of configuring a master root key for the DNS tree. In the proposed fully deployed DNSSEC tree, a master key would be trusted by all domains and used to sign the DNS root keys. The CAIRN master key is designed to imitate the actions of this proposed master root key. The proper use of a master key is still a subject of debate within CAIRN.

5.3.2 Secure Sub-Zones with Unsecure Parent Zones

It is also expected that a CAIRN domain such as sec.nosec.cairn.net may deploy DNSSEC, but the parent nosec.cairn.net may be unsecure. If a sub-domain is unable to obtain KEY set signatures from its parent, the sub-domain may send its KEY set to cairn.net for signing. Secure resolvers should adopt a corresponding policy and agree to trust any zone which is signed by either a parent key or a cairn.net key.

This policy is designed to insure that key chaining can succeed between any two secure domains in CAIRN. The proper approach to handling unsecure parent domains and the overall approach to the DNS key hierarchy is still a subject of debate within CAIRN.

5.4 Trusted Keys and Key Chaining

The CAIRN key signing policies are designed to insure that key chaining will work within the CAIRN portion of the DNS. To obtain a secure DNS record from CAIRN, a resolver should adopt the following policy for learning zone keys:

1) statically configure the CAIRN master key as a trusted key
2) trust any cairn.net KEY with a valid SIG from the CAIRN master
3) trust any domain.cairn.net KEY with a valid SIG from a
   trusted cairn.net KEY
4) trust any sub.domain.cairn.net KEY with either
     a valid SIG from a trusted domain.cairn.net key

or a valid SIG from a trusted cairn.net key

Section 6.3 describes a potential forged KEY attack that can occur if the guidelines are not followed.

CAIRN is interested in inter-operation and key chaining with other portions of the DNS tree. If all sites deployed DNSSEC, one approach could be to trust any root key signed by the DNS master and trust any zone key which is signed by its parent. However, it is unlikely that universal deployment of DNSSEC will be achieved quickly. The current CAIRN key chaining policy attempts to overcome problems due to incremental deployment. This approach clearly does not scale and is presented as starting point for further discussion. CAIRN is currently investigating key chaining between CAIRN and other portions of the DNS tree.

6 Operational Scenarios

This section provides examples scenarios for DNSSEC operations within the CAIRN domain. All the examples assume a resolver requests the A record set for the host hostx.secure.isie.cairn.net. It is assumed the resolver knows how to obtain the proper A records, KEY records, and SIG records. The scenarios describe how these records are used to authenticate the result.

In the first scenario, an unbroken tree of DSA and RSA is assumed. In the second scenario, the tree is broken by a domain which does not implement DNSSEC. The third scenario considers a forged key attack and describes why a resolver must be aware of the CAIRN signing policy. The fourth scenario considers an example where some domains do not implement the RSA algorithm. The fourth scenario is currently unworkable, addressing this problem appears to introduce either security holes or scaling problems related to key signing and resolver rules.

6.1 Operations in an Unbroken Tree.

In the simplest operational scenario, each zone has a KEY set with both an RSA and DSA. Each KEY set is signed by both the RSA and DSA key from the zone's parent. The top level zone KEY set, cairn.net, is signed by the both the RSA and DSA master key. The DSA and RSA master keys are trusted by a resolver.

We assume the resolver has obtained the A record set for hostx.secure.isie.cairn.net and also received the two SIG records, one SIG generated by the DSA key from secure.isie.cairn.net and

one SIG generated by the RSA key from secure.isie.cairn.net. After obtaining the necessary KEY records and corresponding SIGs, the resolver can return either an DSA authenticated answer or an RSA authenticated answer.

If the resolver knew the trusted DSA for secure.isie.cairn.net, the resolver could verify the DSA SIG which covers the A records. The resolver must learn the RSA key for secure.isie.cairn.net.

The resolver obtains the KEY set for secure.isie.cairn.net and its corresponding SIGS. The DSA SIG covering this KEY set indicates that it was generated by the isie.cairn.net DSA KEY. Under CAIRN policies, isie.cairn.net has authority to sign KEY sets for the domain secure.isie.cairn.net. The resolver must now learn the DSA for isie.cairn.net.

The resolver obtains the KEY set for isie.cairn.net and its corresponding SIGS. The DSA SIG covering this KEY set indicates that it was generated by the cairn.net DSA KEY. Under CAIRN policies, cairn.net has authority to sign KEY sets for the domain isie.cairn.net. The resolver must now learn the DSA for cairn.net.

The resolver obtains the KEY set for cairn.net and its corresponding SIGS. The DSA SIG covering this KEY set indicates

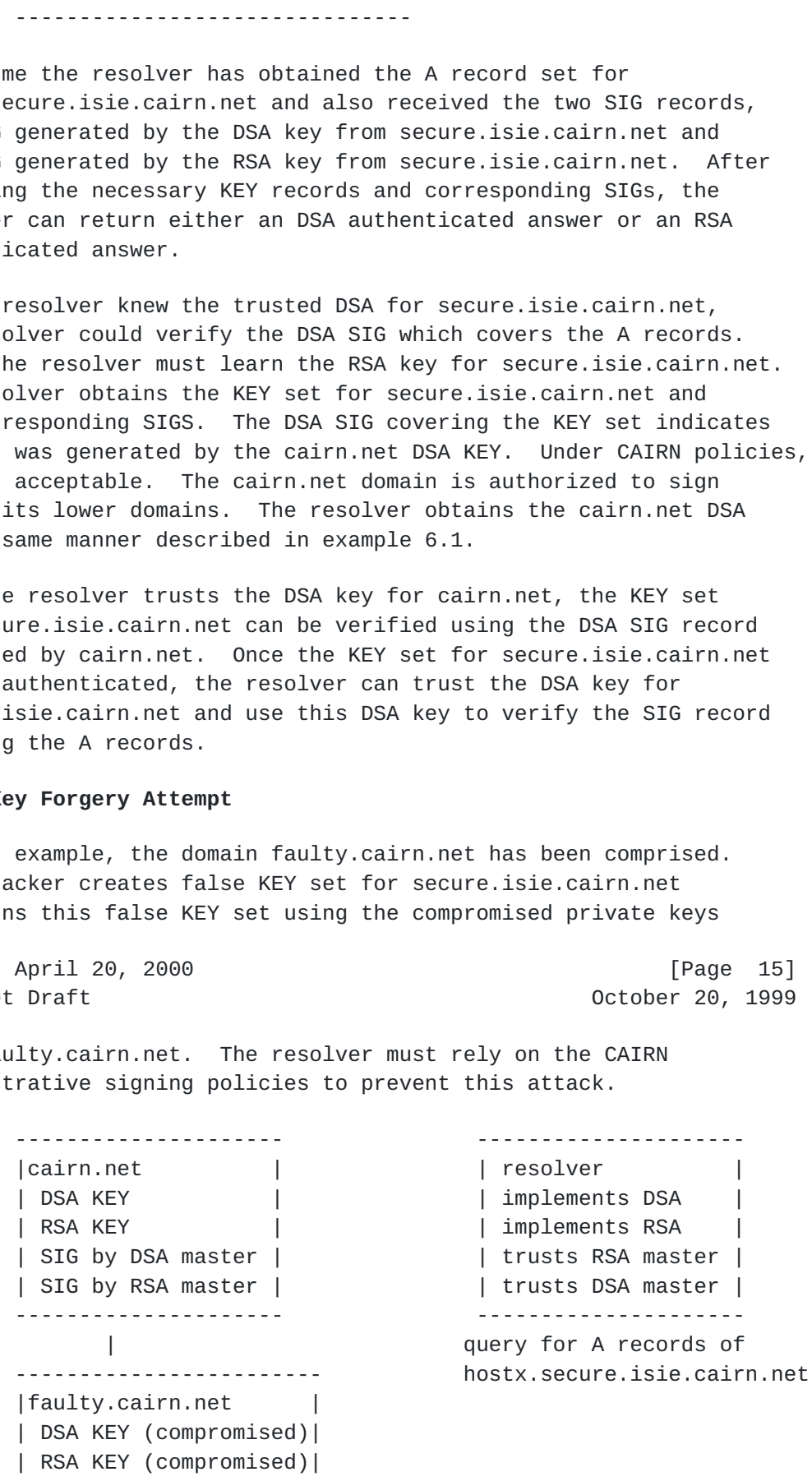
that it was generated by the cairn.net master DSA KEY. Under CAIRN policies, the cairn.net master has authority to sign KEY sets for the domain cairn.net. The resolver is configured to trust the master for key for cairn.net and can no begin checking SIGs.

The resolver uses the trusted master DSA key for cairn.net and checks the DSA SIG record covering the KEY set for cairn.net. Once the KEY set for secure.isie.cairn.net has be authenticated, the resolver can trust the DSA key for cairn.net and use this DSA key to verify the KEY set for isie.cairn.net. The DSA SIG record authenticates the isie.cairn.net KEY set and the trusted DSA KEY for isie.cairn.net is obtained. The isie.cairn.net DSA KEY checks the DSA SIG record covering the secure.isie.cairn.net KEY set and the secure.isie.cairn.net DSA key is obtained. Finally, the secure.isie.cairn.net DSA key is used to verify the DSA SIG covering the A records for hostx.secure.isie.cairn.net.

A similar process allows the resolver to authenticate the A records set using RSA keys and RSA SIG records.

6.2 Operations in an Broken Tree (Unsecured Parent Zone).

In this operational scenario, the cairn.net and secure.isie.cairn.net zones have KEY sets with both an RSA and DSA keys. The isie.cairn.net zone has not deployed DNSSEC and there no KEY records associated with isie.cairn.net. The secure.isie.cairn.net domain will follow the CAIRN guidelines and have its KEY set signed by cairn.net. The DSA and RSA master keys are again trusted by a resolver.



We assume the resolver has obtained the A record set for hostx.secure.isie.cairn.net and also received the two SIG records, one SIG generated by the DSA key from secure.isie.cairn.net and one SIG generated by the RSA key from secure.isie.cairn.net. After obtaining the necessary KEY records and corresponding SIGs, the resolver can return either an DSA authenticated answer or an RSA authenticated answer.

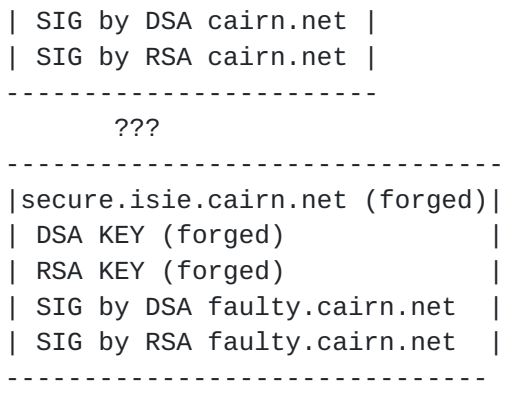
If the resolver knew the trusted DSA for secure.isie.cairn.net, the resolver could verify the DSA SIG which covers the A records. Again the resolver must learn the RSA key for secure.isie.cairn.net. The resolver obtains the KEY set for secure.isie.cairn.net and its corresponding SIGS. The DSA SIG covering the KEY set indicates that it was generated by the cairn.net DSA KEY. Under CAIRN policies, this is acceptable. The cairn.net domain is authorized to sign any of its lower domains. The resolver obtains the cairn.net DSA in the same manner described in example 6.1.

Once the resolver trusts the DSA key for cairn.net, the KEY set for secure.isie.cairn.net can be verified using the DSA SIG record generated by cairn.net. Once the KEY set for secure.isie.cairn.net has be authenticated, the resolver can trust the DSA key for secure.isie.cairn.net and use this DSA key to verify the SIG record covering the A records.

6.3 A Key Forgery Attempt

In this example, the domain faulty.cairn.net has been comprised. The attacker creates false KEY set for secure.isie.cairn.net and signs this false KEY set using the compromised private keys

from faulty.cairn.net. The resolver must rely on the CAIRN administrative signing policies to prevent this attack.



The resolver obtains a false A record set for hostx.secure.isie.cairn.net and also received the two SIG records which cover the A record set. One SIG is generated by the forged DSA key from secure.isie.cairn.net and one SIG generated by the forged RSA key from secure.isie.cairn.net.

The resolver requests the KEY set for secure.isie.cairn.net and receives the forged KEY set from the attacker. The forged KEY set is covered by both a DSA SIG and RSA SIG from the compromised faulty.cairn.net keys. The faulty.cairn.net keys are signed by cairn.net who is in turn signed by the master. Simply following the chain trust will lead the resolver to accept the forged secure.isie.cairn.net keys. The only protection against this attack is CAIRN key signing rules. In CAIRN, a domain's keys can only be signed either the domain's parent or the cairn.net key. According to the CAIRN policy, the domain faulty.cairn.net is not authorized to sign the KEY set for secure.isie.cairn.net.

Note if the attacker has compromised either the cairn.net key or the isie.cairn.net keys, then the CAIRN policy offers not defense and the attacker can forge data from secure.isie.cairn.net.

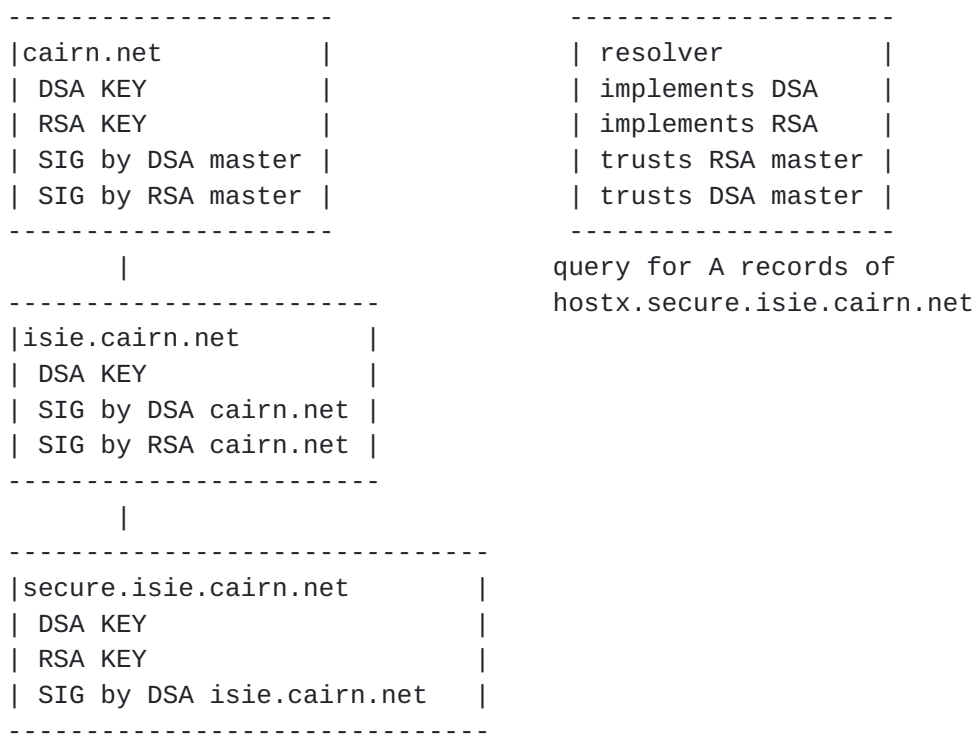
This scenario need not be limited to compromised keys.
For example, it is reasonable for a zone such as netjava.cairn.net

to demand that its competitor, pcjava.cairn.net, is not authorized to sign (and thus forge) its KEY set. Note that java and pcjava are a purely fictional domains and any relationship to any existing malicious companies is purely coincidental.

6.4 Operations in an Mixed Key Algorithm Tree.

In this operational scenario, cairn.net and secure.isie.cairn.net have KEY sets with DSA and RSA keys. However, isie.cairn.net has chosen not to use an RSA key and instead generates a NULL RSA KEY. Since there is no RSA KEY at isie.cairn.net, only a DSA SIG record

covers the secure.isie.cairn.net KEY set. The DSA and RSA master keys are again trusted by a resolver.



We assume the resolver has obtained the A record set for hostx.secure.isie.cairn.net and also received the two SIG records, one SIG generated by the DSA key from secure.isie.cairn.net and one SIG generated by the RSA key from secure.isie.cairn.net. After obtaining the necessary KEY records and corresponding SIGs, the resolver can return either an DSA authenticated answer as described in example scenario 6.1. Depending on how algorithm inter-operation is defined, the resolver may or may not be able to to use the RSA SIG from secure.isie.cairn.net.

If the resolver knew the trusted RSA for secure.isie.cairn.net, the resolver could use the RSA SIG record to authenticate the A record set. Again the resolver must learn the RSA key for secure.isie.cairn.net, but this time the resolver has no RSA SIG record which covers the secure.isie.cairn.net KEY set. The resolver could use the DSA SIG record to authenticate the KEY set and learn the RSA KEY for secure.isie.cairn.net, but this approach can

be problematic. It may be that an attacker has exploited some vulnerability in the DSA algorithm. The attacker then uses this vulnerability to forge a false RSA key for secure.isie.cairn.net. The "RSA authenticated" result returned the resolver could have been compromised by a vulnerability in the DSA algorithm.

Reporting this result as authenticated by RSA would be misleading at best.

The resolver could instead report that there is no RSA SIG covering the secure.isie.cairn.net KEY set and thus no RSA authenticated answer is available. Before doing this, the resolver should first authenticate the fact that no RSA SIG exist for secure.isie.cairn.net. However, this means the secure.isie.cairn.net domain received better security when its parent was unsecure. In scenario 6.2, the isie.cairn.net was unsecure and the KEY set for secure.isie.cairn.net was signed by both the RSA and DSA keys from cairn.net. This approach argues for per algorithm DNS key hierarchies, further complicating KEY set signing and resolver rules for which KEYs to trust.

7 IMPLEMENTATION IN THE GENERAL INTERNET

There seem to be several main areas that need to be addressed to facilitate the implementation of DNSSEC on a wider scale:

i) Administrative and configuration activities associated with key distribution and signing
ii) Compatibility with older versions of BIND, especially those which are acting as secondary servers to a master server which may be processing DNSSEC related resource records. (Some older version of BIND may have trouble when receiving SIG and KEY records)
iii) Definition of how application resolvers should interact with the secure name servers in order to determine the security status of DNS records. Options include extending application resolvers to perform authentication processing, utilizing transaction signatures or IPSec between application resolvers and secure name servers, or relying on site security and inspection of query response flags to determine security status.

Item i above, seems to be the most difficult issue when contemplating the transition from non-secure to secure DNS operations. The DNSSEC operational concept includes the following ideas relating to key distribution and signing:

- There is a public master key at the root level for which all name servers have knowledge of and configure as a trusted key in their configuration files.
- Every DNS zone administrator generates their own keys and signs their zone files. In addition, a DNS zone administrator must send its keys to its parent zone so that the parent can sign the sub-zones keys. The result is that every zone has in its zone file a copy of its keys along with a signature of its

keys by its parent keys. In this manner, if there are no breaks in the parent-child signing process going up the DNS hierarchy, it will be possible for any name server to get and verify any key that might need to verify a DNS resource record.

The following problems can be readily identified relative to the transition to secure DNS operations.

1. The root servers will probably not be the first name servers ready to take on the task of generating keys, signing zone files, and signing children zone keys.
2. The procedures for the parent-child key signing exchange are not well understood, especially for a parent node which has a lot of delegation points. Automation techniques need to be developed.
3. It must be anticipated that there will be gaps in the parent-signing key signing operations as one moves up from different spots in the DNS hierarchy. This will prevent successful authentication in some cases.

The following mechanism is being investigated as a vehicle to allow organizations to immediately begin using DNSSEC. The basic approach is to allow secure sub-zone to be placed anywhere in the current DNS hierarchy by allowing the parent-child key signing hierarchy to deviate from the general DNS hierarchy. This would allow secure sub-zones to authenticate data from any other secure sub-zone which also participates in this program.

As a vehicle to test this concept, the CAIRN TOC is planning to provide top level key signing services for other secure sub-zones whose natural parent are not doing so. In this manner all secure sub-zones which had their keys signed by the CAIRN master keys, would only need to configure the public portion of the CAIRN master key in their name server configuration in order to have the ability to use key chaining to authenticate each others records.

It is envisioned that this would provide an infrastructure for the DNSSEC community to work on some of the operational, automation, and administrative issues. The ability of the CAIRN TOC to provide these services would be dependent on the number of participants as well as the general workload required for these activities.

The steps for an organization to create a secure sub-zone would be as follows:

1. Identify the zone which will be secure.
2. Install DNSSEC capable BIND, configure with the CAIRN master key as the trusted key.
3. Generate keys and sign zone files.

4.Provide CAIRN TOC with the keys have been generated.
5.CAIRN will act as the top level key signing authority,
and will sign the keys, and send back copies of the SIG records.

This concept is currently under test with the following secure sub-zone:
secure.cs.ucla.edu

8 SUMMARY AND FUTURE EXPERIMENTATION AREAS

DNSSEC implementation and experimentation activities in CAIRN are expected to continue. A key concern and area of interest for this activity is further satisfaction of the implementation objectives stated in Section 2.2 of this draft. In summary they are:

- Minimize the administrative burden imposed on a parent zone resulting from a child zone changing their configuration.

It is desired that the zone file requirements delegating a zone under DNSSEC be the same as it is for current DNS operations. That is the delegation point only needs to specify the delegated sub-zone and the name servers which serve that sub-zone. In this manner, child sub-zones will not cause a parent zone to have to modify and resign its zone file when the sub-zone changes its secure status, generates new keys, or makes other changes to its configuration which would not require a parent zone file modification under traditional DNS implementations.

The signature of the child zone keys by the parent keys is expected to be an off-line activity, which will only impact the zone file for the child zone requesting the signature services.

Minimization of the impact to the parent zone file is considered very important, especially for zones with many delegation points.

-Provide mechanisms which allow for non-contiguous secure zones space.

In order to allow for non-contiguous secure zones space, the CAIRN implementation will allow the key signing hierarchy to deviate from the general DNS zone hierarchy. That is, if a secure sub-zone has a parent sub-zone which is unsecure, then that sub-zone can submit its keys to the cairn.net zone administrators for signature. The intent is that all name servers which have the cairn.net master key configured as a trusted key will then be able to authenticate records from other zones which do the same.

This requires an administrative policy which states that the cairn.net master key is "authorized" to sign records from other zones, which

are not directly delegated from the cairn.net zone. This concept of associating a trusted key with a list of zones it is authorized to sign for it not part of the current DNSSEC operational considerations. Discussion of the need (and mechanisms) to create associations between trusted keys and authorization for zone signing is desired.

It is felt by the authors of this draft, that initial implementation in the general Internet DNS hierarchy will require that pockets of secure DNS zones be able to exist. There must be a mechanism which allows name servers and application resolvers which are adequately configured to obtain and authenticate DNS records.

Other DNSSEC areas interest for which CAIRN research and experimentation is anticipated includes:

-automation of the key generation and zone file signing.
-automation of the submission of a key to a parent for signature
-mechanisms to allow a non-secure application resolver to use a secure name server to obtain authenticated DNS records. Evaluation of the use of IP Security (IPsec) vs Transaction Signatures is needed. Providing mechanisms for applications to inspect DNS response flags and record contents to make intelligent decision about data security is also needed.

9 IANA CONSIDERATIONS

This document does not place any requirements on the assigned numbers authority.

10 SECURITY CONSIDERATIONS

This entire document is a note on security considerations.

11 AUTHOR'S ADDRESS

Daniel Massey
<masseyd@cs.ucla.edu>
Computer Science Department
UCLA
Los Angeles, CA 90095
+1 (703) 599-7318

Tom Lehman
<tlehman@isi.edu>
4350 North Fairfax Drive
Suite 770
Arlington, VA 22203
+1 (703) 812-3736

Edward Lewis
<lewis@tislabs.com>
3060 Washington Rd. (Rte 97)
Glenwood, MD 21738
+1 (443) 259-2352

12 REFERENCES

This section will be more formally defined as the document progresses.

11 FULL COPYRIGHT STATEMENT